

Clase 4 - Requerimientos de ambientes ágiles- user stories

viernes, 23 de agosto de 2024 21:20

Hay una cuestión relacionada a la forma en que desarrollamos software el cual llamamos proceso de desarrollo de software... hoy en día convive dos tipos de procesos, los empíricos y los definidos.

Esta diferencia o clasificación que armamos ahora, es el hilo conductor de todos los temas que veremos en la materia todo tiene un punto de vista en los procesos definidos y empíricos

No se discute la necesidad de hacer gestión de configuración.

Un ejemplo de procesos definidos sería el PUD/RUP, tienen como características, que hay una concepción organizacional que define una forma de trabajar, y espera que todos los proyectos que se lleven en adelante respeten esa definición de proceso que se planteó...

Proceso(se instancia) *se adapta* -> proyecto (gráfico de proceso, proyecto y producto)

Con el proceso establecemos todo lo que hay que para hacer software: requerimientos, análisis, diseño... despliegue.

La org define lo máximo que puede llegar a necesitar que es la intensidad del PUD y del RUP y cada vez que inicia un proyecto el equipo adapta el proceso al proyecto al particular teniendo en cuenta distintas circunstancias... la gente que necesito, las características del producto, etc.. Muchas consideraciones que me ayudan a determinar que de todo lo que está escrito en el proceso voy a necesitar.

Hay un conjunto de decisiones que no son del equipo que vienen determinadas por la organización.

Todo definido para que sean predecibles... eso busca el enfoque de los procesos definidos -> poder estimar el comportamiento de un equipo en función de comportamientos pasados.

Cada vez que haces software la salida es distinta a igual entrada nunca tengo igual salida... no importa si es definido.

La mala implementación de este proceso generó rechazo a hacer procesos y en los 2000 surge la corriente agilismo

¿qué es el agilismo?

No es una metodología ni un proceso... es una filosofía y pensamiento.... Apunta a la parte de la cultura de las personas que lo adoptan

Ágil es una ideología con un conjunto definido de principios que guían el desarrollo del producto

Valores de los equipos ágiles ...

- Planificación continua, multi-nivel
- Facultados, auto-organizados, equipos completos
- Entregas frecuentes, iterativas y priorizadas
- Prácticas de ingeniería disciplinadas
- Integración continua
- Testing Concurrente

EL manifiesto ágil da el inicio formal del agilismo... creando como un estándar

Ágil no es velocidad... es falso que no hace falta documentar...

El agilismo o pensamiento ágil usa procesos empíricos para funcionar... se basan en la experiencia del mismo equipo... van ganando experiencia(con aciertos y errores).

- Balance entre ningún proceso y demasiado proceso. La diferencia inmediata es la exigencia de una menor cantidad de documentación, sin embargo no es eso lo más importante:

- Los métodos ágiles son adaptables en lugar de predictivos.
- Los métodos ágiles son orientados a la gente en lugar de orientados al proceso.

El agilismo sostiene que la experiencia no es extrapolable

En los procesos empíricos empoderan al equipo... las decisiones que antes tomaban organizacionalmente y las bajaban a los equipos ahora las toman los que van a hacer el trabajo.

Los procesos empíricos **tienen 3 pilares**:

Los p. empíricos necesitan ciclo de vida iterativo

- Inspección
- Adaptación
- Transparencia

EL empirismo funciona con ciclo de vida iterativo con iteraciones lo más cortos que el equipo considere oportuna (scrum recomienda 1 mes) y de ciclos de inspección y adaptación. Acá es donde se genera la experiencia, en determinados intervalos de tiempo paro y reviso lo que estoy haciendo y me adapto.

Necesito inspeccionar y adaptar 2 cosas la forma de que estoy trabajando (proceso, se hace con el equipo) y el producto (se hace con el cliente)

El tercer pilar dice que para que funcione todo necesitamos ser transparente... lo que generamos es del equipo.. El código no es mío.. Si necesito colaboración la pido.... Lo que yo hago o no hago perjudica al equipo.. No ser orgullosos y pedir ayuda de ser necesario.

Cualquier framework o método ágil que quiera adherir al manifiesto necesita primero adherir al proceso empírico.

Los 3 pilares son de los procesos empíricos... el agilismo los adopta porque dicen que ellos trabajan con procesos empíricos.

Manifiesto ágil

Tiene 4 valores y 12 principios

☐ LEER TODOS A PROFUNDIDAD



Los 12 principios del Manifiesto Ágil

- 1 Nuestra mayor prioridad es **satisfacer al cliente.**
- 2 **Aceptar** que los requisitos **cambien.**
- 3 **Entregar** software funcional **frecuentemente.**
- 4 Los responsables de negocios, diseñadores y desarrolladores deben **trabajar juntos** día a día durante el proyecto.
- 5 Desarrollamos proyectos en torno a **individuos motivados.**
- 6 El método más eficiente de comunicar información es **conversaciones cara a cara.**
- 7 El **software funcionando** es la principal **medida de éxito.**
- 8 Los procesos ágiles promueven el **desarrollo sostenible.**
- 9 La **atención continua** a la **excelencia técnica** y al **buen diseño** mejor la Agilidad.
- 10 La **simplicidad** es esencial.
- 11 Las mejores arquitecturas, requisitos, y diseños emergen de **equipos auto-organizados.**
- 12 A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo y de acuerdo a esto **ajustan su comportamiento.**

EL termino para referirnos al cliente es el Product Owner... persona que viene del lado del negocio... referente del cliente... conoce el producto que necesita y el negocio

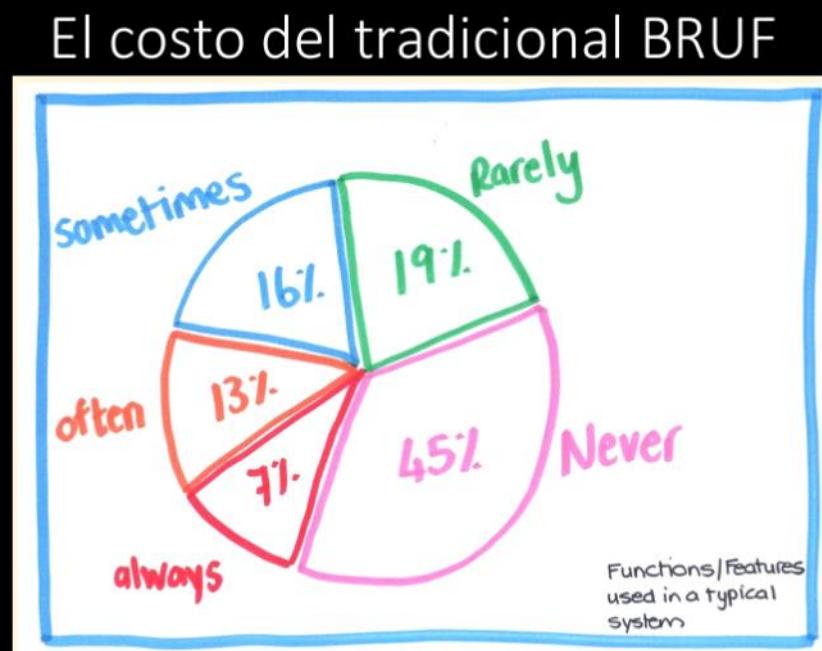
Los técnicos somos nosotros(analistas, desarrolladoras, etc)

Hay contratos pero son distintos a una ERS

Entender el concepto de requerimientos ágiles.. Y user stories que es la técnica que gestiona requerimientos en ambientes ágiles

Los requerimientos hablan de una definidicio de requerimientos JUST IN TIME... voy a trabajar en los requerimientos solo cuando lo necesite.. No invierto tiempo en grandes definiciones.

Los productos "Exitosos" también tienen un desperdicio significativo



Fuente: Jim Johnson of the Standish Group, Keynote Speech XP 2002

Requerimientos en Agile

Los req ágiles implica la definición de requerimientos just in time(ayuda a evitar el desperdicio) no toda la definición del producto de antemano. Diferir las decisiones lo más que se pueda, trabajar con reqs emergente... implica abrazar la idea de que el mejor medio de comunicación

es cara a cara... equipo formado con gente del negocio y tecnica.
Comenzar con un conjunto pequeño de requerimientos y aplicar lo de inspección y adaptación para ir recolectando los requerimientos según los voy necesitando.. Y como es un proceso empírico que tanta documentación formal yo voy a necesitar es algo que el equipo decide en función de las circunstancias del proyecto...



Los **requerimientos ágiles** son una forma flexible y colaborativa de definir lo que se necesita en un proyecto de software, adaptándose a cambios y priorizando lo que genera más valor para el cliente. A diferencia de los enfoques tradicionales, donde los requisitos se documentan al inicio y son rígidos, en el desarrollo ágil los requerimientos se construyen de manera incremental y evolutiva, permitiendo ajustes continuos a medida que el proyecto avanza.

El rol del product owner viene del negocio.. Es el representante de un grupo de personas que están relacionadas con el producto... un producto tiene un PO el cual debe tener capacidad de toma de decisiones dentro del negocio.

USER STORIES

Son requerimientos de usuario, requerimientos a nivel de granularidad del negocio.. Si quisiera compararlos con los CU.. Los CU son requerimientos de software y la user de de usuario o de negocio.. El nivel de abstracción e la users es más alto.

Se las define como una descripción corta de una funcionalidad del producto... describen una necesidad... y quien necesita? El product owner.

En scrum adopta las user pero no es algo nativo de scrum... son una recomendación pero eventualmente puedo usar otra cosa para identificar requerimientos.

Las user stories las tiene que escribir el PO... no es importante quien las escribe si no las ideas.. Las ideas son del product owner..

Hay un lugar en los ambientes ágiles que es un contenedor de todas las necesidades y requerimientos llamada product backlog ... que es una lista **priorizada** de características del producto... un contenedor de items de product backlog

Dentro de los que puedo encontrar dentro de un PB como items estan las US... pero puede haber otras cosas

US, defectos, deuda técnica(por ejem utilización de una base de datos...), epica(US muy grande), temas, (agrupador de US y epicas), SPIKE

En un product backlog nunca hay tareas.

Las spike asociamos a incertidumbre.. Cuando un nivel de incertidumbre sobre una necesidad

del PO... que nos e cuanto tiempo voy a tardar.. Q que tenego que hacer para construir algo... cuando tenemos un nivel de incertidumbre muy alto aparece la SPIKE.. Que es investigación es una necesidad de profundizar sobre lo que el PO me está pidiendo

Las spikes tienen dos fuentes...

la funcional

la técnica

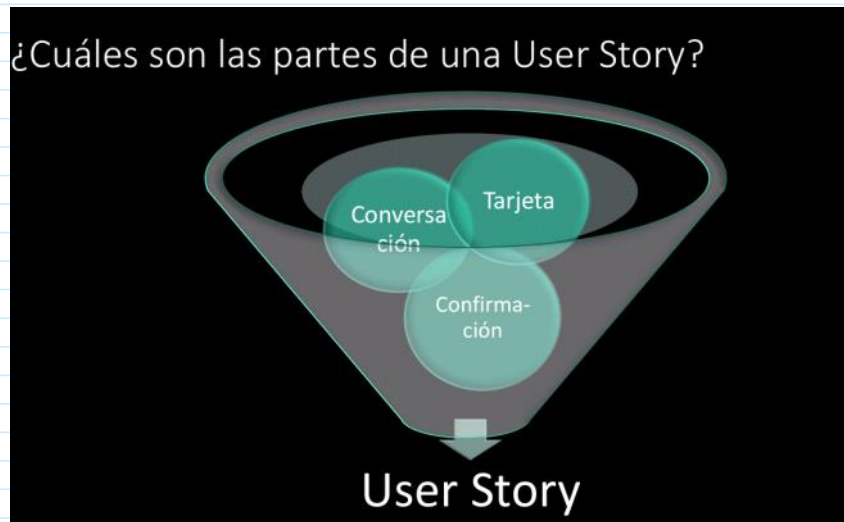
Tamaño de equipos definidor por scrum? 7+-2

Las US tienen 3 partes.. LAS 2 C

Card: parte visible de la US lo que escribimos e ingresamos en el PB.. Hoy usamos software... JIRA... devops...team foundation.. Softwares que permiten gestionar proyectos utilizando como unidad de gestion de requerimientos a las US.

Conversation(la más importante): no queda en ningún lugar... es lo que charlamos nosotros y en base a eso sale la card y al dorso al confirmacion que son las pruebas de aceptación.

Confirmation:



En agilismo se trabaja con gestión binaria (o esta o no esta hecho el trabajo)... por eso la recomendación es que sean chiquitas las US...

LAS US se traducen a CU o a SW dependiendo del equipo ... los equipos maduros no suelen hacerlo a SW.

No hay una obligación de que la US la baje a CU. La decisión la tomara el equipo.

Sintaxis de la tarjeta:

Forma de expresar las Historias de Usuario

As who, I want what so that why.

Como **<nombre del rol>**,
yo puedo **<actividad>**
de forma tal que **<valor
de negocio que
recibo>**

Representa quién está realizando la acción o quién recibe el valor de la actividad.

Representa la acción que realizará el sistema

Comunica porque es necesaria la actividad

COMO <ROL> QUIERO <FUNCIONALIDAD> PARA <VALOR>

User Story: Tarjeta y Pruebas de Aceptación

Buscar Destino por Dirección

Como Conductor quiero buscar un destino a partir de una calle y altura para poder llegar al lugar deseado sin perderme.

Criterios de Aceptación:

- La altura de la calle es un número.
- La búsqueda no puede demorar más de 30 segundos.

Pruebas de Usuario

- Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura existente (pasa).
- Probar buscar un destino en un país y ciudad existentes, de una calle inexistente (falla).
- Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura inexistente (falla).
- Probar buscar un destino en un país inexistente (falla).
- Probar buscar un destino en País existente, ciudad inexistente (falla).
- Probar buscar un destino en un país y ciudad existentes, de una calle existente y demora más de 30 segundos (falla).

Al escribirlo así tengo en claro el valor de negocio que produce entonces es más fácil priorizarlo para el PO

Un ROL no una persona no un cargo organizacional... ese rol lo pueden desempeñar o una persona o más.

El framework que vamos a utilizar (SCRUM) dice que el tamaño de la US debe ser consumible en una iteración (la iteración como max 1 mes.).. Las estimaciones en el agilismo son relativas.

INVEST MODEL: modelo de calidad que se usa para evaluar las US que escribimos..

- INDEPENDIENTE
- NEGOCIABLE: el product owner te dice que necesita el cómo lo hace la parte técnica.
- VALOR DE NEGOCIO
- ESTIMABLE
- SMALL
- TESTEABLE